Lab 6: Concurrent SMs

### Introduction

There is only one exercise for this lab. The objective of this lab is to introduce you to an example of concurrent SMs, as well as introduce you to one of the complexities you may want to use in your final project: the 4-digit 7 segment display.

### Hardware Specifications

**Inputs**

* 1x Button (Joystick button is acceptable)

**Outputs**

* 1x 4-digit 7 segment display (must use at least two of the four digits)
* 5-7 LEDs

**Additional Components**

* Arduino
* Breadboard
* Jumper Cables

### Software Specifications

**Design Requirements**

* Use, at least, two Concurrent SMs (can have the same period).
* Please use the Timer.h header file as done in previous labs. (No delay)

**Expected Behavior**

\**Sample Demo for Lab 6:* <https://drive.google.com/file/d/1nP8yVBrMxivR4Cs5ChJMeoNg2gDuuNTg/view?usp=share_link>

*\*This lab can be broken into two subprojects. One is a SM that controls the LED behavior. And another SM to control the 4-digit 7 segment display.*

***LED Behavior:***

* Recommended Period: 10ms
* When powered, only 1 LED will ever be lit at a given instant.
* When the button is not pressed, the led that is lit will rapidly shift from right to left, then back to right, then left, back again… (see sample demo above).
* When the button is pressed, the led currently on has been selected and will remain on. No other leds will turn on.
* When the button is released the led will resume shifting rapidly from right to left…

***4-digit 7 Segment Display Behavior:***

* + Recommended Period: 10ms
  + The 4-digit 7 segment display keeps track of the number of times the green led was consecutively selected.
  + Every time the red led is selected, the counter resets to 0.
  + The highest score you should be able to achieve (technically speaking) is 99. (So make sure you’re using two digits).

*\*Even though you may not reach a score higher than 9 in your demo, we still ask you to display 2 digits as the code for displaying one digit vs two is very different.*

### Submission Guidelines

**Video Demo Guidelines**

*\*When demoing this lab, you are welcome to increase the period to demonstrate the functionality.*

*\*The highest score from Tuesday’s discussions was 22 (not me). The period was 10ms.*

* Please show the following scenarios in your video:
  + Button pressed on the wrong LED (i.e. one that doesn’t increment the score).
  + Button pressed on the correct LED (i.e. one that increments the score).
  + Then try to win a couple of points and hit the wrong LED to show the score reset on the 7 segment display.

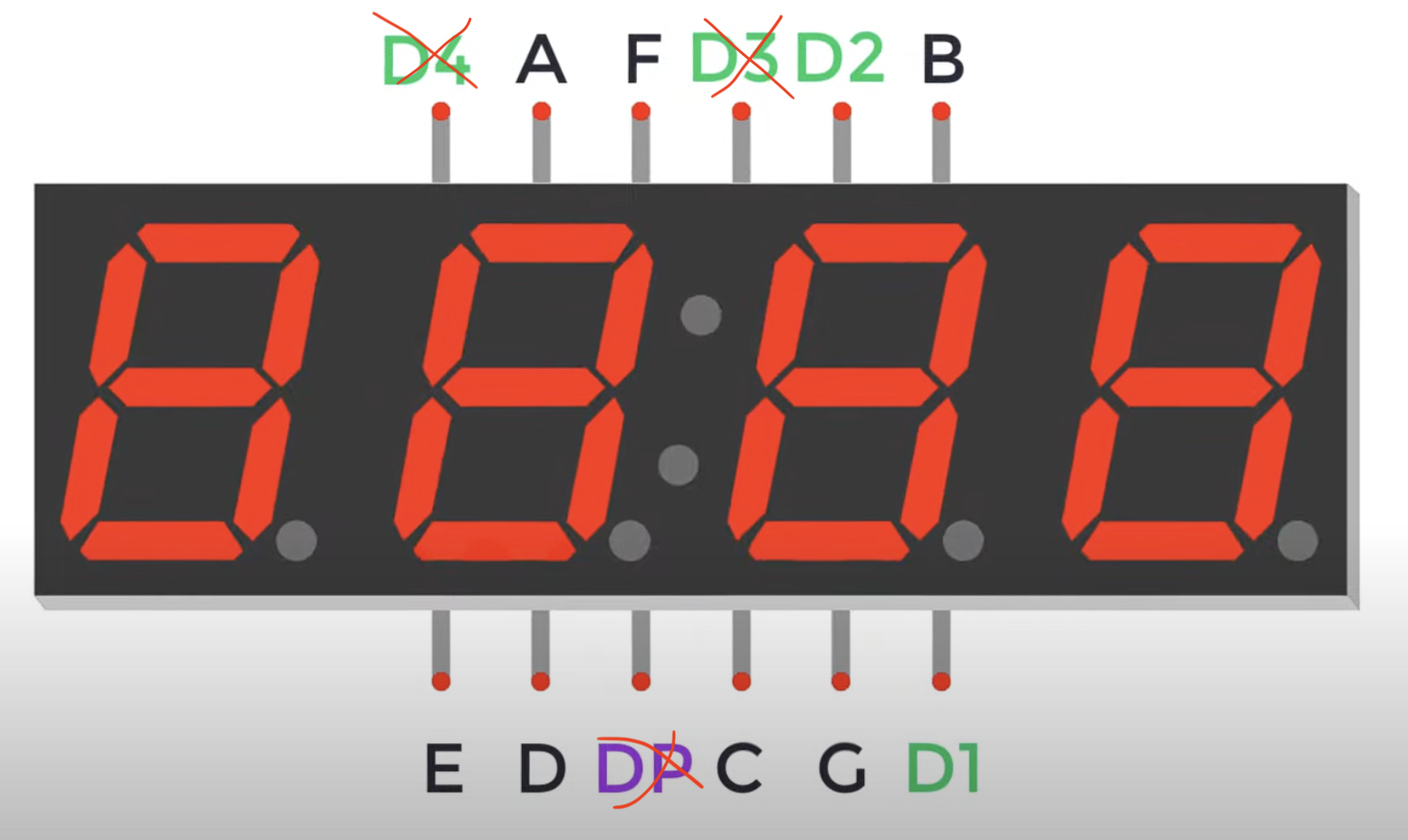
**Gradescope Submission Guidelines**

1. Create a separate YouTube video of your lab and provide a pdf file with the link to your video.
2. Please show your face at the beginning of the video for a second or two and then move over to the board without editing.
3. Also, please show the input and output clearly in the video so that we can grade you properly.
4. Turn in your code as a separate .cpp file on Gradescope.
5. Submit all .cpp and .pdf files under one submission on Gradescope. A total of 2 files should be submitted (1 .cpp and 1 .pdf).

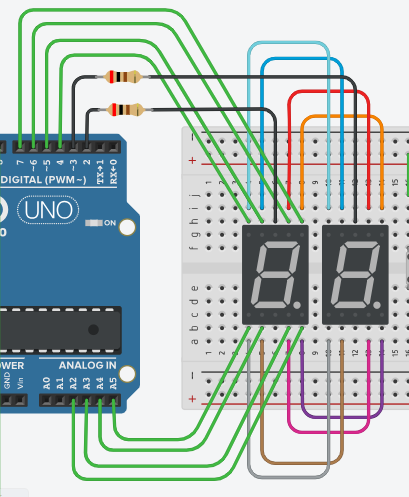
Do not submit to Zybooks.

### Additional Resources

* Sample Demo (on Drive): <https://drive.google.com/file/d/1nP8yVBrMxivR4Cs5ChJMeoNg2gDuuNTg/view?usp=share_link>
* How a 4-digit 7 seg display works: <https://www.youtube.com/watch?v=fYAlE1u5rno>
* The Pinout for the diagram on TinkerCAD is NOT the same as the pinout for the 4-digit 7 segment display you have in your kit. Please refer to the following diagram to see the pinout for the 4 digit 7 segment display.



* ***Note: You do not need to connect D4, D3, and DP.***
* We briefly touched upon the pinout in discussion. This was brought up as it tells us an important functionality we’ll be using for this lab: the analog pins A0-A5 can function as digital pins. You can use them the same way as you use the digital pins. For example, pinMode(A0, OUTPUT); digitalWrite(A0, LOW); is a valid way to set pin A0 to digital low.
* Many of you were interested in a sample implementation of a concurrent SM. I’ve provided one in the same folder for Lab 6. Keep in mind this implementation is done in C and for a different microcontroller so some of the implementation specifics will be different. The contents of scheduleTasks is what should go in your void loop() function.
* I’ve also provided a helper function for writing one number to one digit of the hex display. Check the Lab 6 Folder for the helper function code.



* ONLY FOR THOSE USING TINKER CAD: The 4-digit 7seg display we are provided in our lab kits is not available on TinkerCAD. To build a similar circuit for 2 digits (which is all that is required for this lab) is provided above.
  + Things to note about the above diagram:
    - The resistors are 220 Ohm.
    - When wiring the hardware, you need to wire ONLY the black and green wires. All other wires are done internally (within the hardware 4-digit 7 segment display).
    - There are two “classes” of wires you need to worry about:
      * The digit wires (Black wires)
        + Control entire digits.
        + Is powered/enabled/on when LOW
      * The segment wires (Green wires)
        + Control individual segments of a digit.
        + Is powered/enabled/on when HIGH.